Beowulf Database Architecture

Strawman design 1

Dec. 5, 2002

Bob Schaefer

Design Criterion

- Operational
  - Ingest new/updated files without undue interruption
  - Large data staging area (2 Tb?)
  - Quick data transfers between nodes
  - Concurrent searching

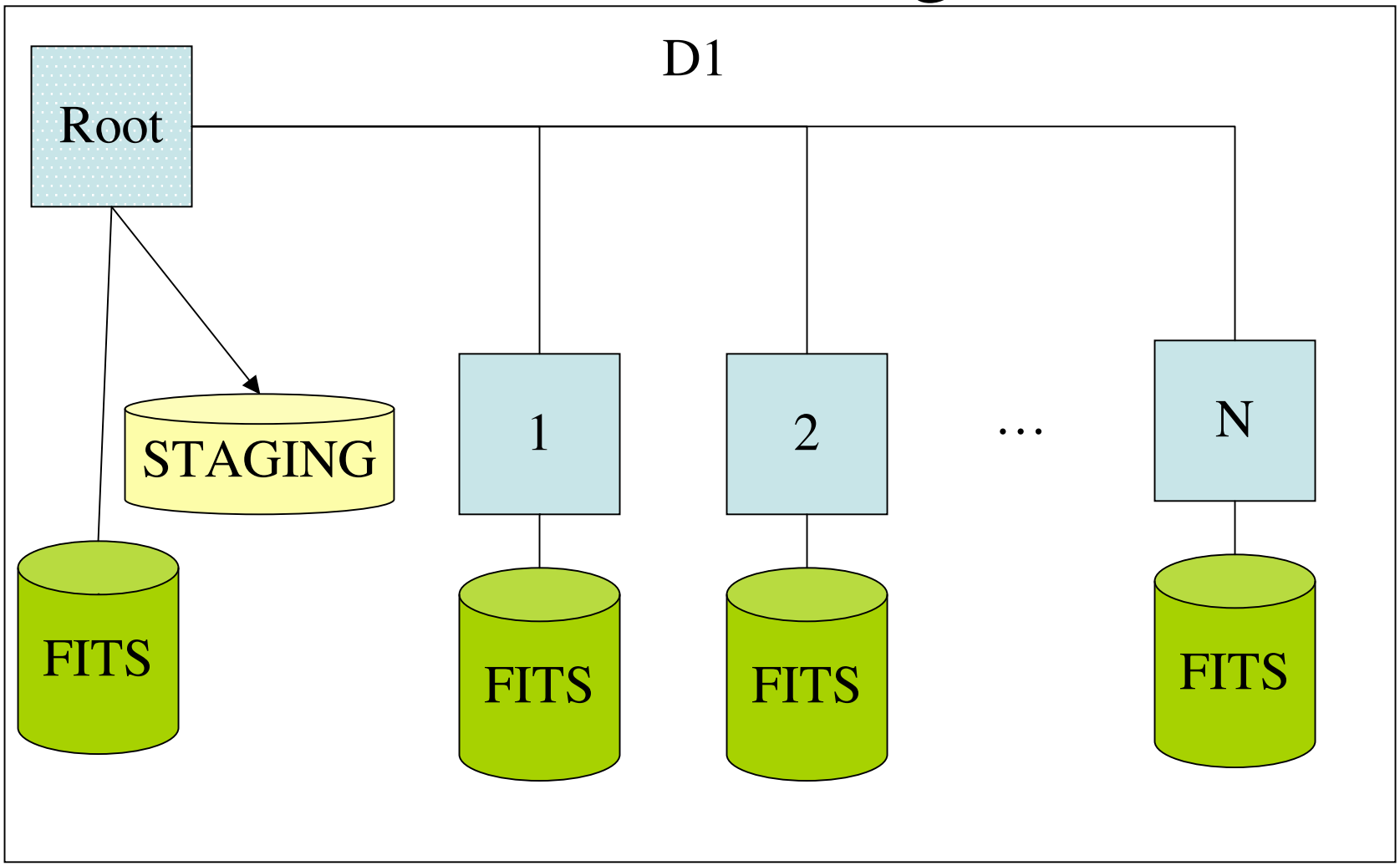
Design Criteria 2

- Reliability
  - Handle node failures (crashes)
  - Handle disk failures
  - Automatic verification of data file versions
  - Handle main (root) server failures without too much interuption

Design Criteria 3

- External communication
  - Want cluster to handle queries only from a single host (not required, but I like the cleanness of this choice)
  - 1 Beowulf server to be middleman between outside world (web interface, ftp service, and ingest from LIOC pipeline
  - Want logging of queries and results outside of Beowulf. (easier for Sys admins to maintain).

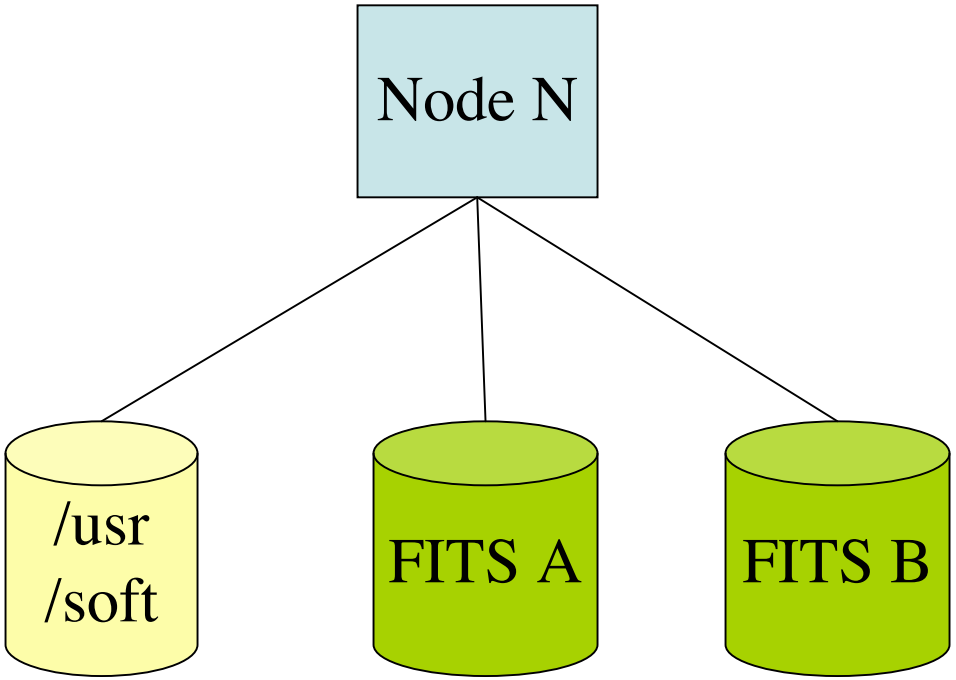
Cluster design



Cluster design notes

- Disks
  - FITS Disks on nodes contain all FITS files with all photons and are mirrored on the node. I.e., 2 (~200Gb) disks are locally mounted on each node. If one disk fails the other one can then be used immediately. (this allows automated switchover, but is this needed?)
  - Staging disk(s) is(are) mounted by root node(s) and an external node that is ftp accessible.
  - A third disk would contain the operating system and software

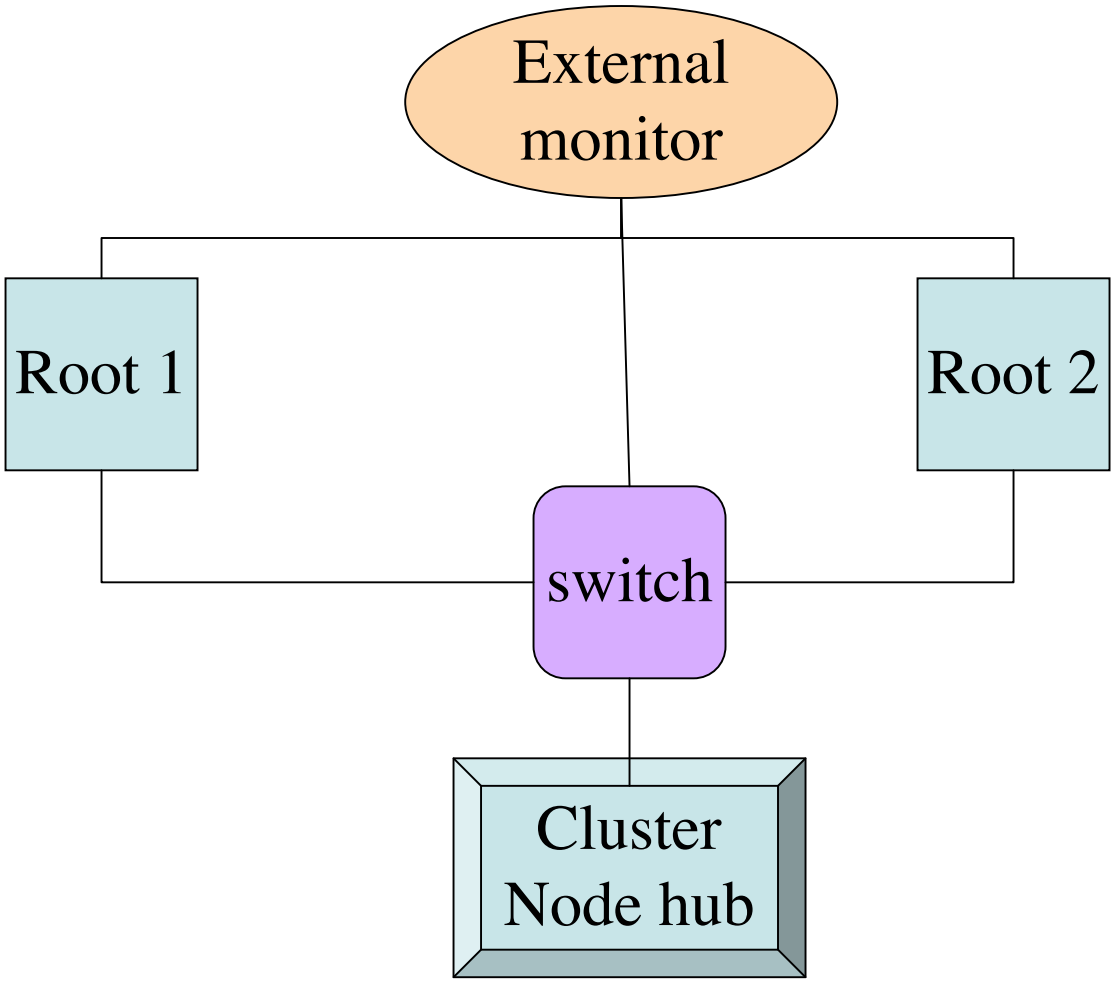
Disk Mounts



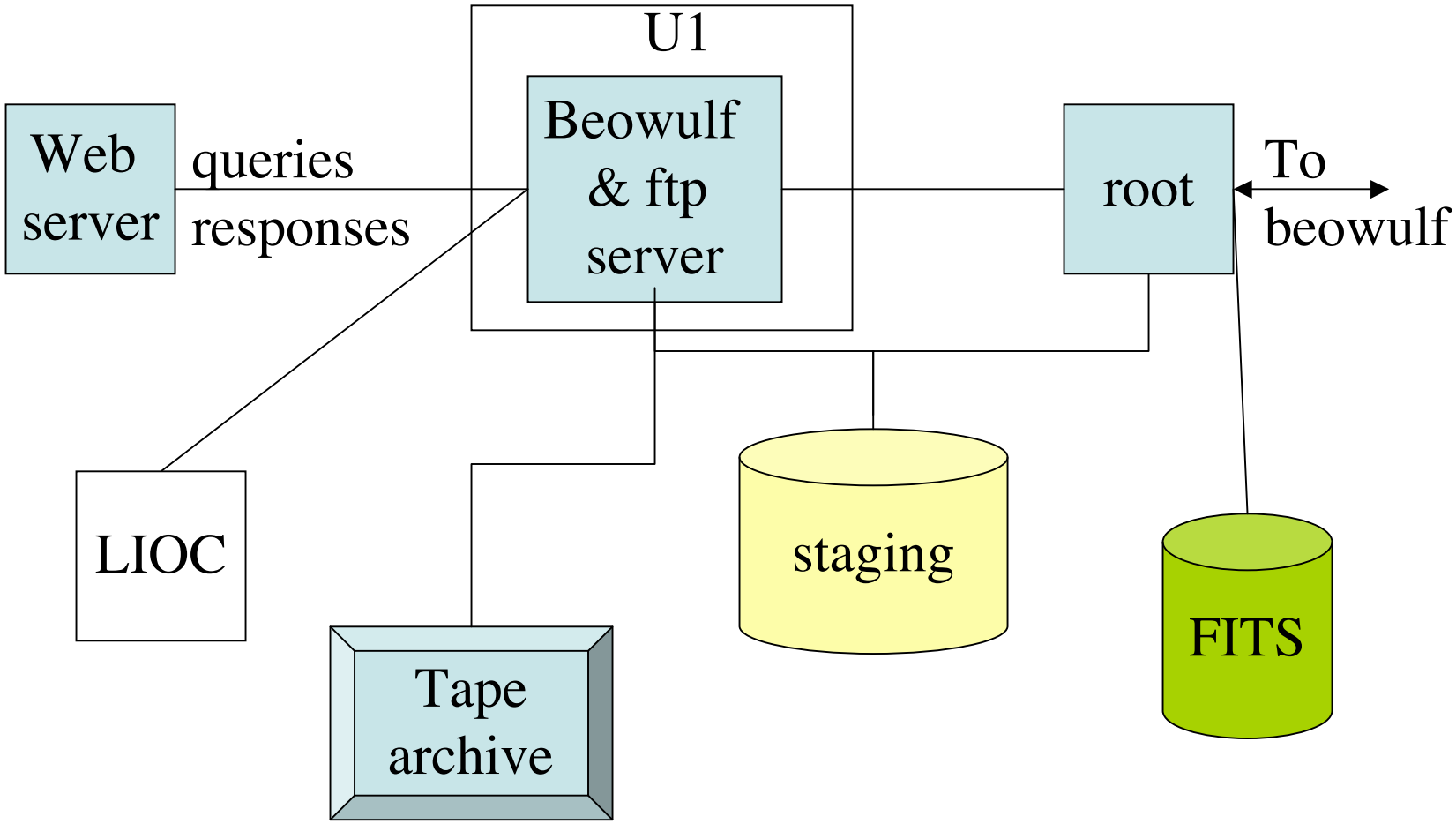
Cluster Design notes 2

- Nodes
  - Each node has 3 disks drivers (SCSI?)
  - Root note has 2 ethernet cards: 1 to talk to outside and 1 on beowulf net. All other nodes have 1 ethernet card.
  - Root node is mirrored (copy is switched out of the network until needed). Not sure how to do this.

Root Node mirroring



External Connections



Cluster Processes

- Ingest Pipeline (1st 3 steps on Beowulf server)
  - LIOC pipeline produces new/reprocessed FITS file.
  - Filter Program reads file defining photons, then creates new FITS file with only photons.
  - FITS file verification+checksum generated.
  - New photon FITS file sent to root node
  - New file copied to all nodes
  - FITS index file with filenames, time spans, checksums and file sizes updated. Index file copied to all nodes.
  - Beowulf search server process (on root) told to re-read index file.

Database agents

- Each node has a process which periodically checks FITS index file against disk contents. (file sizes, names and checksums)
- Each node checks disk operational status
- Root node checks state of slave nodes (and itself).

External communication

- External Request Queue
  - Requests put on to-do list until completed and logged. (light database?)
  - Beowulf server has a db which tracks requests (request criteria, searcher id, query status, query completion status, query result ftp status.)
  - Beowulf root sends out query status (queued, processing, complete + path to results)

External Communication 2

- Ingest communication
  - New data file notification to ROOT
  - File transfer status to Beowulf server
- Error messages (out of Root node)
  - Processing failures
  - Hardware failures.

Staging Disk contents

- FITS files resulting from queries in separate subdirectories.
- FITS files ready for ingest (+ checksum and file size information)
- Log files for Beowulf search engine.